CLAIMS

What is claimed is:

1. A computer-implemented method comprising:

receiving a data cipher operation; and

processing the data cipher operation, wherein the processing comprises generating a number of portions of ciphertext from plaintext, wherein a load operation associated with the generating of at least one portion of the ciphertext executes prior to a store operation associated with the generating of a prior portion of the ciphertext.

2. The computer-implemented method of claim 1, wherein the generating of the at least one portion of the ciphertext and the generating of the prior portion of the ciphertext is executed within one iteration of a number of iterations for the data cipher operation.

3. The computer-implemented method of claim 2, wherein the generating of the at least one portion of the ciphertext is re-executed in a iteration that is subsequent to the one iteration upon determining that data retrieved from the load operation conflicts with data stored in the store operation.

4. The computer-implemented method of claim 1, wherein the store operation comprises swapping data within a data structure, the data within the data structure used in generating the ciphertext.

5. The computer-implemented method of claim 4, wherein the load operation comprises accessing data from the data structure.

6. The computer-implemented method of claim 5, wherein the generating of the at least one portion of the ciphertext is aborted upon determining that the data being swapped equals the data being accessed in the data structure.

7. The computer-implemented method of claim 5, wherein the data cipher operation comprises an RC4 operation and wherein the data structure comprises a substitution-box.

8. A computer-implemented method executing in a processor, the method comprising:

receiving a request to perform for data ciphering of plaintext; and

processing the request based on a data structure stored in a memory coupled to the processor, wherein the processing comprises,

performing a first access of data from the data structure;

swapping the data from the first access;

data ciphering a first portion of the plaintext based on the swapped data from the first access;

performing a second access of data from the data structure prior to the swapping of the data from the first access; and

performing the following, upon determining that the data from the first access does not equal the data from the second access,

swapping the data from the second access; and

data ciphering a second portion of the plaintext based on the swapped data from the second access.

9. The computer-implemented method of claim 8, wherein the processing of the request is executed within one iteration of a number of iterations.

10. The computer-implemented method of claim 9, comprising performing the following, upon determining that the data from the first access equals data from the second access:

reexecuting the performing of the second access of data from the data structure;

swapping the data from the second access; and

data ciphering the second portion of the plaintext based on the swapped data from the second access.

11. The computer-implemented method of claim 8, wherein the data ciphering comprises an RC4 operation.

12. The computer-implemented method of claim 8, wherein the data structure comprises a substitution-box.

13. An apparatus comprising:

a memory to store a data structure; and

a processing unit coupled to the memory, the processing unit to execute a data ciphering operation, wherein the processing unit is to swap data stored in the data structure for data ciphering of a first portion of plaintext, and wherein, prior to the completion of the swapping of the data stored in the data structure for data ciphering of the first portion of the plaintext, the processing unit is to access data stored in the data structure for data ciphering of a second portion of the plaintext.

14. The apparatus of claim 13, wherein the processing unit is to data cipher the second portion of the plaintext upon determining that the data being swapped in the data structure does not equal the data being accessed in the data structure.

15. The apparatus of claim 13, wherein the processing unit is to execute the data ciphering operation across a number of iterations, wherein the swapping of data stored in the data structure for data ciphering of the first portion of plaintext and the accessing of data stored in the data structure for data ciphering of the second portion of the plaintext are executed within one iteration of the number of iterations.

16. The apparatus of claim 15, wherein the processing unit is to reexecute, within a subsequent iteration of the number of iterations, the accessing of data stored in the data structure for data ciphering of the second portion of the plaintext, upon determining that the data swapped for data ciphering of the first portion of plaintext equals the data accessed for the data ciphering of the second portion of the plaintext.

17. The apparatus of claim 13, wherein the memory is to store the plaintext.

18. The apparatus of claim 13, wherein the data ciphering operation comprises an RC4 operation.

19. The apparatus of claim 13, wherein the data structure comprises a substitution-box.

20. The apparatus of claim 13, wherein the apparatus is coupled to a host processor and a host memory, wherein the processing unit is to receive the data ciphering operation from the host memory.

21. A co-processor coupled to a host processor and a host memory, the co-processor comprising:

an interface unit to retrieve a data encryption operation, a substitution (S)-box and plaintext associated with the data encryption operation from the host memory based on an instruction from the host processor; and

an execution unit coupled to the interface unit, the execution unit comprising,

a memory to store the plaintext and the S-box associated with the operation for the data cipher;

a microcontroller unit to schedule the data cipher operation; and

a RC4 unit to receive the data cipher operation, wherein the RC4 unit is to swap data stored in the S-box for data ciphering of a first portion of the plaintext and wherein the RC4 unit is to read data stored in the S-box for data ciphering of a second portion of the plaintext, prior to completion of the swapping of data stored in the S-box for data ciphering of the first portion of the plaintext.

22. The co-processor of claim 21, wherein the RC4 unit is to data cipher the second portion of the plaintext upon determining that the data being swapped in the S-box does not equal the data being read from the S-box.

23. The co-processor of claim 21, wherein the RC4 unit is to data cipher the first portion of the plaintext.

24. The co-processor of claim 21, wherein the RC4 unit is to swap data retrieved from the S-box for the data ciphering of the second portion of the plaintext upon determining that the data being swapped for the data ciphering of the first portion of the plaintext does not equal the data read from the S-box for data ciphering of the second portion of the plaintext.

25. An apparatus comprising:

a memory to store a substitution (S)-box;

an RC4 hardware state machine coupled to the memory to generate a plurality of output text blocks from a plurality of input text blocks, wherein a subset of said plurality of output text blocks are generated as a result of repeating the same sequence of states, wherein during each of the repeated sequence of states data is speculatively read from said S-box in said memory as part of the generation of a next one of said plurality of output text blocks prior to a write to said S-box in said memory completing as part of generation of a current one of said plurality of output text blocks.

26. The apparatus of claim 25, wherein said plurality of output text blocks are ciphertext blocks and said plurality of input text blocks are plaintext blocks.

27. The apparatus of claim 25, wherein said plurality of input text blocks are ciphertext blocks and said plurality of output text blocks are plaintext blocks.

28. A system comprising:

a host processor;

a host memory coupled to the host processor, the host memory to include a security operation, wherein the security operation includes a data cipher operation based on RC4, the host memory to include plaintext and a data structure for the data cipher operation;

a co-processor coupled to the host processor, the co-processor comprising,

an interface unit to retrieve the security operation from the host memory based on an instruction from the host processor;

an execution unit coupled to the interface unit, the execution unit comprising,

a memory to store the plaintext and the data structure associated with the data cipher operation;

a microcontroller unit to store the data cipher operation in an execution queue; and

an RC4 unit coupled to the execution queue, the RC4 unit to receive the data cipher operation, wherein the RC4 unit is to swap data stored in the S-box for data ciphering of a first portion of the plaintext and wherein the RC4 unit is to read data stored in the S-box for data ciphering of a second portion of the plaintext, prior to completion of the swapping of data stored in the S-box for data ciphering of the first portion of the plaintext.

29. The system of claim 28, wherein the RC4 unit is to data cipher the second portion of the plaintext upon determining that the data being swapped in the data structure does not equal the data being read from the data structure.

30. The system of claim 28, wherein the RC4 unit is to data cipher the first portion of the plaintext.

31. The system of claim 28, wherein the RC4 unit is to swap data retrieved from the data structure for the data ciphering of the second portion of the plaintext upon determining that the data being swapped for the data ciphering of the first portion of the plaintext does not equal the data read from the data structure for data ciphering of the second portion of the plaintext.

32. A machine-readable medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

receiving a data cipher operation; and

processing the data cipher operation, wherein the processing comprises generating a number of portions of ciphertext from plaintext, wherein a load operation associated with the generating of at least one portion of the ciphertext executes prior to a store operation associated with the generating of a prior portion of the ciphertext.

33. The machine-readable medium of claim 32, wherein the generating of the at least one portion of the ciphertext and the generating of the prior portion of the ciphertext is executed within one iteration of a number of iterations for the data cipher operation.

34. The machine-readable medium of claim 33, wherein the generating of the at least one portion of the ciphertext is re-executed in a iteration that is subsequent to the one

iteration upon determining that data retrieved from the load operation conflicts with data stored in the store operation.

35. The machine-readable medium of claim 32, wherein the store operation comprises swapping data within a data structure, the data within the data structure used in generating the ciphertext.

36. The machine-readable medium of claim 35, wherein the load operation comprises accessing data from the data structure.

37. The machine-readable medium of claim 36, wherein the generating of the at least one portion of the ciphertext is aborted upon determining that the data being swapped equals the data being accessed in the data structure.

38. The machine-readable medium of claim 36, wherein the data cipher operation comprises an RC4 operation and wherein the data structure comprises a substitution-box.

39. A machine-readable medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

receiving a request to perform data ciphering of plaintext; and

processing the request based on a data structure stored in a memory coupled to the processor, wherein the processing comprises,

performing a first access of data from the data structure;

swapping the data from the first access;

data ciphering a first portion of the plaintext based on the swapped data from the first access;

performing a second access of data from the data structure prior to the swapping of the data from the first access; and

performing the following, upon determining that the data from the first access does not equal the data from the second access,

swapping the data from the second access; and

data ciphering a second portion of the plaintext based on the swapped data from the second access.

40. The machine-readable medium of claim 39, wherein the processing of the request is executed within one iteration of a number of iterations.

41. The machine-readable medium of claim 40, comprising performing the following, upon determining that the data from the first access equals data from the second access:

reexecuting the performing of the second access of data from the data structure;

swapping the data from the second access; and

data ciphering the second portion of the plaintext based on the swapped data from the second access.

42. The machine-readable medium of claim 39, wherein the data ciphering comprises an RC4 operation.

43. The machine-readable medium of claim 39, wherein the data structure comprises a substitution-box.

44. The machine-readable medium of claim 39, wherein processing the request for data ciphering of the plaintext comprises data ciphering the plaintext over a number of iterations and wherein the data ciphering of the first portion of the plaintext is in a same iteration as the data ciphering of the second portion of the plaintext.